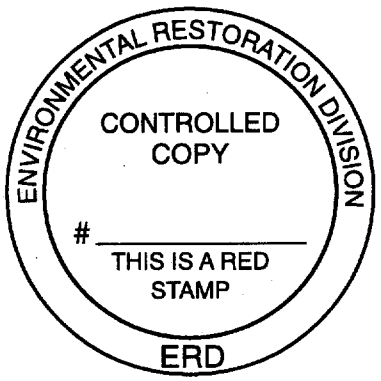
LLNL Environmental Restoration Division (ERD)
Standard Operating Procedure (SOP)

ERD SOP 5.10: Data Management Receipt and Processing by Electronic Transfer—Revision: 2



AUTHOR(S):
L. Graves and N. Montez

APPROVALS:	Date
Albert L Lamarre **Division Leader**	9/8/00
Carol Stoker **Information Systems Management Group Leader**	9/8/00

CONCURRENCE:	Date
Valerie Dibley **QA Implementation Coordinator**	9/5/00

*CMG Contract Services

1.0 PURPOSE

The purpose of this procedure is to establish the means for receiving and processing lithology data received by electronic transfer from the Hydrogeology Group (HG) of Environmental Restoration Division (ERD). This procedure is to ensure complete and consistent handling of electronically transferred lithology data within the Data Management Team (DMT) of the ERD Information System Management Group (ISMG).

2.0 APPLICABILITY

This procedure applies to personnel receiving and processing electronically transferred lithology data.

3.0 REFERENCES

Not applicable.

4.0 DEFINITIONS

See SOP Glossary.

5.0 RESPONSIBILITIES

5.1 ERD Data Management Team (DMT)

The DMT's responsibilities are to receive lithologic data and process them as outlined in this procedure and communicate with the ERD HG geologists regarding necessary corrections or clarifications of reported lithologic data.

5.2 ERD Hydrogeology Group (HG)

The ERD HG is responsible for creating ASCII files of relevant lithologic data and delivering the files to DMT.

6.0 PROCEDURE

6.1 Transfer of data.

6.1.1 Log onto EPDBS.

6.1.2 Transfer files to the home directory being used.

6.1.3 Start a log file that records all work done by typing 'touch *filename*.rcd' at the prompt. Save all results from the following operations done in OpenIngres to the file.

6.2 Verify Content of ASCII file

6.2.1 File content should have the following format:

borehole name|top depth|bottom depth|lithology

6.2.2 Execute chkchar to verify that file contains no illegal characters.

%chkchar lith6aug97.asc > chklith

6.2.3 Use 'more chklith' to look at the files for the bad characters, especially control Z, control M, %, and *.

6.3 Transfer to Working Table

6.3.1 To transfer into an OpenIngres personal working table, wlithologic, execute SQL command, which is best done from isql (%isql epddata) using the lithin.txt sq1 or the following SQL:

copy table wlithologic(loc_id='c0|', intvltop='c0|', intvlbot='c0|', lithology='c0') from '/home/*username/filename'*;

6.4 Verify and Complete Contents of Working Table

6.4.1 Verify number of rows in wlithologic with number of rows in ASCII file.

select count(*) from wlithologic;

6.4.2 Verify that borehole names are standard location codes included in the location and boreholesprg tables. Correct any bad names in the wlithologic table. If corrections need to be made in the location table and boreholesprg table, consult another DMT member. Use lithlocchk.sq1 or the following SQL:

```
select distinct loc_id

from wlithologic

where loc_id not in

(select l.id from location l);

select distinct loc_id

from wlithologic

where loc_id not in

(select l.loc_id from boreholesprg l);
```

6.4.3 Enter entered date, status=draft, and source={contractor who prepared ASCII files} using SQL. If needed, left-hand justify any alpha numeric data. Use the lithupd.txt or the following SQL:

```
update wlithologic

set status='D',source='WEISS';

update wlithologic

set entered = '07-aug-1997'

where entered='';
```

6.4.4 Check to see if any sets of lithologies start with depths other than 0.0. Confirm any necessary changes with well logs or the person who converted the well logs to electronic files. Use lithminchk.txt or the following SQL:

```
select distinct loc_id,min(depth)

from wlithologic

group by loc_id;
```

6.4.5 Check for duplicate rows within the work table using lithinterdup2.txt or by the following SQL:

```
select a.loc_id,a.intvltop,a.tid,b.tid,a.lithology
```

from wlithologic a,wlithologic b

where a.loc_id=b.loc_id

and a.depth=b.depth

and a.lithology=b.lithology

and a.tid>b.tid;

6.4.6 Remove duplicate rows from the work table by executing the following SQL:

/*delete from wlithologic

where tid in

(select a.tid from wlithologic a,wlithologic b

where a.loc_id=b.loc_id

and a.intvltop=b.intvltop

and a.tid>b.tid);

select a.loc_id,a.intvltop,a.tid,b.tid,a.lithology,b.lithology

from wlithologic a,wlithologic b

where a.loc_id=b.loc_id

and a.intvltop=b.intvltop

and a.tid>b.tid;*/

Remove the '/*' and '*/' when duplicates are confirmed and ready to be deleted and rerun.

6.4.7 Check for duplicates between the work table and global lithologic table using lithdup.txt or executing the following SQL:

select distinct loc_id

from wlithologic

where

loc_id in

(select loc_id from lithologic);

6.4.8 Remove duplicates by executing the following SQL:

```
/*delete from wlithologic

where loc_id in

 (select loc_id from lithologic);

select distinct loc_id

from wlithologic

where

loc_id in

 (select loc_id from lithologic);*/
```

Remove the '/*' and '*/' when duplicates are confirmed and ready to be deleted and rerun.

6.5 Append to Global Lithologic Table

6.5.1 Append by executing the following SQL command: (lithupload.txt)

```
select count(*) from wlithologic;

select count(*) from lithologic;

insert into lithologic(loc_id,intvltop,intvlbot,lithology,entered,

examined,comment,status,source)

select loc_id,intvltop,intvlbot,lithology,entered,examined,

comment,status,source

from wlithologic;

select count(*) from wlithologic;

select count(*) from lithologic;
```

6.5.2 Verify that the count in lithologic has increased by the correct number of rows and check in lithologic table to verify that desired lithology was truly appended.

6.5.3 Delete rows from wlithologic.

6.5.4 Make entry into Global Updates Logbook of rows added.

6.5.5 File documentation in lithologic global appends file.

7.0 QA RECORDS

7.1 Global Updates Logbook

7.2 Lithology table in EPDData

8.0 ATTACHMENTS

Not applicable.